Sequential Copying Networks

Qingyu Zhou\textsuperscript{1}, Nan Yang\textsuperscript{2}, Furu Wei\textsuperscript{2}, Ming Zhou\textsuperscript{2}
\textsuperscript{1}Harbin Institute of Technology, Harbin, China \textsuperscript{2}Microsoft Research, Beijing, China
qyzhgm@gmail.com \{nanya, fuwei, mingzhou\}@microsoft.com

* Work done during an internship at Microsoft Research.

Why Sequential Copying?

Abstractive summarization as an example:

Sequential Copying happens when doing abstractive summarization:

- Copying a span from input sentence

An example of sequential copying in abstractive sentence summarization task.

Sequential Copying is ESSENTIAL for such tasks and datasets

Sequential Copying Networks (SeqCopyNet)

Problem: “Single word copying” paradigm may introduce errors due to separate copying decisions when copying multi-words

Solution: We copy them once and for all

Copy Switch Gate

Span Pointer Network

Copy State Transducer

SeqCopyNet can copy long spans from input sentence

SeqCopyNet is good at detecting boundaries:
- named entities
- noun phrases

After copying a long span, the decoder can still generate well

Conclusion

- SeqCopyNet enables multi-word span copying, and can be integrated with seq2seq framework
- SeqCopyNet is good at detecting boundaries, such as named entity
- We release a new abstractive sentence summarization test set
- Future work: apply SeqCopyNet to other tasks such as dialogue generation

Experiments

Task 1: Question Generation give a sentence and its desired answer

Dataset: Question Generation dataset based on SQuAD.

Evaluation Metric: BLEU-4

<table>
<thead>
<tr>
<th>Model</th>
<th>Dev set</th>
<th>Test set</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCFG-Trans\textsuperscript{1}</td>
<td>9.28</td>
<td>9.31</td>
</tr>
<tr>
<td>s2s-attn\textsuperscript{1}</td>
<td>3.01</td>
<td>3.06</td>
</tr>
<tr>
<td>NQG\textsuperscript{1}</td>
<td>10.06</td>
<td>10.13</td>
</tr>
<tr>
<td>NQG+ (single copy)</td>
<td>12.30</td>
<td>12.18</td>
</tr>
<tr>
<td>SeqCopyNet</td>
<td>13.13</td>
<td>13.02</td>
</tr>
</tbody>
</table>

Task 2: Abstractive Sentence Summarization

Dataset: English Gigaword: Rush, Chopra, and Weston (2015)*, Zhou et al. (2017b) and our internal test sets

Evaluation Metric: ROUGE F1

<table>
<thead>
<tr>
<th>Models</th>
<th>Test set in Zhou et al. (2017b)</th>
<th>Our internal test set</th>
</tr>
</thead>
<tbody>
<tr>
<td>RG-1</td>
<td>RG-2</td>
<td>RG-L</td>
</tr>
<tr>
<td>ABS\textsuperscript{1}</td>
<td>37.41</td>
<td>15.87</td>
</tr>
<tr>
<td>s2s-attn (greedy)</td>
<td>46.21</td>
<td>24.02</td>
</tr>
<tr>
<td>NMT + UNK_PS (greedy)</td>
<td>45.64</td>
<td>23.38</td>
</tr>
<tr>
<td>NMT + UNK_PS (beam)</td>
<td>47.05</td>
<td>24.82</td>
</tr>
<tr>
<td>SEASS greedy\textsuperscript{1}</td>
<td>45.27</td>
<td>22.88</td>
</tr>
<tr>
<td>SEASS beam\textsuperscript{1}</td>
<td>46.86</td>
<td>24.58</td>
</tr>
<tr>
<td>SeqCopyNet (greedy)</td>
<td>46.51</td>
<td>21.44</td>
</tr>
<tr>
<td>SeqCopyNet (beam)</td>
<td>47.27</td>
<td>25.07</td>
</tr>
</tbody>
</table>

\* Test sets of Rush, Chopra, and Weston (2015) and Zhou et al. (2017b) are similar, so we only show results on Zhou et al. (2017b) in this poster.

Example Output

SeqCopyNet can copy long spans from input sentence

SeqCopyNet is good at detecting boundaries:
- named entities
- noun phrases

Conclusion

- SeqCopyNet enables multi-word span copying, and can be integrated with seq2seq framework
- SeqCopyNet is good at detecting boundaries, such as named entity
- We release a new abstractive sentence summarization test set
- Future work: apply SeqCopyNet to other tasks such as dialogue generation